UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/675,116	09/28/2000	Grzegorz J. Czajkowski	SUN-P5075-RSH	9136

22835　　7590　　05/21/2004		EXAMINER
PARK, VAUGHAN & FLEMING LLP		ALI, SYED J
508 SECOND STREET		

ART UNIT	PAPER NUMBER
2127	9

PARK, VAUGHAN & FLEMING LLP
508 SECOND STREET
SUITE 201
DAVIS, CA 95616

DATE MAILED: 05/21/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.		Applicant(s)
Office Action Summary	09/675,116		CZAJKOWSKI ET AL.
	Examiner		Art Unit
	Syed J Ali		2127

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed
  after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any
  earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 April 2004*.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-11,13-18 and 20-24* is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-11,13-18 and 20-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some *   c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage
application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1.   A request for continued examination under 37 CFR 1.114, including the fee set forth in

37 CFR 1.17(e), was filed in this application after final rejection.   Since this application is

eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e)

has been timely paid, the finality of the previous Office action has been withdrawn pursuant to

37 CFR 1.114. Applicant's submission filed on April 16, 2004 has been entered.

2.   This office action is in response to the amendment filed April 16, 2004. Claims 1-4, 6-

11, 13-18, and 20-24 are presented for examination.

3.   The text of those sections of Title 35, U.S. code not included in this office action can be

found in a prior office action.

Claim Objections

4.   **Claims 1, 8, and 15 are objected to because of the following informalities:**

a.   In line 21 of claim 1, "argument" should read "arguments".

b.   In line 22 of claim 8, "argument" should read "arguments".

c.   In line 24 of claim 15, "argument" should read "arguments".

Appropriate correction is required.

Claim Rejections - 35 USC § 103

5.   Claims 1-4, 6-11, 13-18, and 20-24 are rejected under 35 U.S.C. 103(a) as being

unpatentable over Gosling (USPN 5,668,999) in view of Jagannathan et al. (USPN

6,496,871) (hereinafter Jagannathan).

6.   As per claim 1, Gosling teaches the invention as claimed, including a method for

verifying type safety of an application snapshot, the method comprising:

the application snapshot includes a subprogram, an operand stack, and a point of

execution (col. 6 lines 28-46);

examining the application snapshot to identify the subprogram and the point of execution

within the subprogram (col. 7 lines 31-44);

examining the subprogram to determine an expected structure of the operand stack at the

point of execution (col. 7 line 10-65);

validating that the state of the application snapshot on the second computing device is

consistent with the expected structure of the operand stack (col. 7 line 58-65);

verifying that variables and arguments within the application snapshot are of the proper

type (col. 10 lines 16-55); and

if the state of the application snapshot is validated as consistent with the expected

structure of the operand stack, executing the application snapshot on the second computing

device (col. 10 lines 56-64).

7.   Jagannathan teaches the invention as claimed, including the following limitations not

shown by Gosling:

the application snapshot including a state of an executing program that is moved from a first computing device to a second computing device across a network in order to continue execution on the second computing device (col. 20 lines 45-55);

receiving the application snapshot of the executing program from the first computing device on the second computing device (col. 18 lines 4-23);

resuming execution of the application snapshot at the point of execution on the first computing device (col. 18 lines 4-23).

8.   It would have been obvious to one of ordinary skill in the art to combine Gosling and Jagannathan since Gosling, while providing means for verifying an application before execution, fails to specify how migration of an application might be handled. Rather, the procedure for verifying an application snapshot is presented, but an assumption is made that once the program begins executing, it will continue to reside on that machine. With the advent of mobile code and distributed processing, a method of handling process migration during process execution is necessary. Additionally, since many networks are heterogeneous, verification across platforms is necessary. Jagannathan provides a system that allows dynamic process migration while also maintaining state information related to the ongoing execution. Jagannathan acknowledges that the problem of migration of ongoing processes has been addressed, but the prior art does not allow for state information related to those processes to be easily migrated (col. 4 lines 47-59). Jagannathan also indicates that process migration may occur between heterogeneous machines, indicating a specific need for a verification method, such as the one disclosed by Gosling (col. 18 lines 4-23). Jagannathan seeks to improve the prior art by not only providing a system that allows migration of ongoing processes, but also allows the state information to be migrated (col.

5 lines 28-35). The combination of Gosling and Jagannathan would provide an exemplary model for verifying the type safety of an executing application, while maintaining state information related to the application as it is migrated from one machine to another.

9.   As per claim 2, Gosling teaches the invention as claimed, including the method of claim 1, wherein examining the subprogram to determine the expected structure of the operand stack at the point of execution involves examining the subprogram with a code verifier, wherein the code verifier ensures that:

    the subprogram does not cause the operand stack to overflow and underflow (col. 8 line 46 - col. 9 line 18);

    a use of a local variable does not violate type safety (col. 10 lines 16-27); and

    an argument of an instruction is of an expected type (col. 6 lines 6-13).

10.   As per claim 3, Gosling teaches the invention as claimed, including the method of claim 1, wherein the operand stack contains at least one local variable, at least one argument that is passed as a parameter to the subprogram, and an offset to the point of execution within the subprogram (col. 5 lines 21-29; col. 6 lines 6-13; col. 6 lines 28-46).

11.   As per claim 4, Gosling teaches the invention as claimed, including the method of claim 2, wherein the expected structure of the operand stack includes a collective size of entries and the types of entries expected on the operand stack at the point of execution within the subprogram (col. 7 lines 20-30).

12.   As per claim 6, Gosling teaches the invention as claimed, including the method of claim 4, wherein validating that the state of the application snapshot on the second computing device is consistent with the expected structure of the operand stack involves ensuring that the collective size of entries and the types of entries on the operand stack agree with the collective size of entries and the types of entries expected on the operand stack (col. 9 lines 44-62).

13.   As per claim 7, Jagannathan teaches the invention as claimed, including the method of claim 1, wherein resuming execution of the application snapshot involves restarting the subprogram at the point of execution within the second computing device (col. 18 lines 4-23).

14.   As per claim 22, Jagannathan teaches the invention as claimed, including the method of claim 1, further comprising restoring the state of an object within the application snapshot on the second computing device by changing a pointer from an address of the object on the first computing device to an address of the object on the second computing device (col. 21 lines 7-27).

15.   As per claims 8-11, 13-14, and 23, Gosling teaches the invention as claimed, including a computer-readable storage medium storing instructions that when executed by a computer causes the computer to perform the method of claims 1-4, 6-7, and 22, respectively (Fig. 2).

16.   As per claim 15-18, 20-21, and 24, Gosling teaches the invention as claimed, including

an apparatus that implements the method of claims 1-4, 6-7, and 22, respectively (Fig. 2).

Response to Arguments

17.   Applicant's arguments filed April 16, 2004 have been fully considered but they are not

persuasive.

18.   Applicant argues on pages 10-11, *"verifying a bytecode program [which is not executing]*

is not the same as verifying a snapshot of an executing program." Applicant thus concludes,

"the system in Gosling cannot 'validate that each variable within objects 206, arguments 218,

and local variables 220 is of the proper type '".

19.   Examiner respectfully disagrees. While Gosling may not expressly recite taking a

snapshot of an executing program and then migrating the program, Gosling does allow for

verification of an executing program, including validation of all variables, arguments, and other

data associated with the application. Gosling expressly states that the bytecode program verifier

may temporarily store stack information for an executing program to verify its type safety (col. 2

lines 8-24). Gosling specifically verifies local variables and verifies the type codes for safety

(col. 10 lines 16-55).

20.   Applicant argues on page 10, *"There is nothing within Gosling or Jagannathan, either*

explicit or implicit, which suggests forming a snapshot of an executing program on a first

computing device, transferring the snapshot to a second computing device, and verifying on the

second computing device that the snapshot is consistent with the point of execution of the

program on the first computing device. Furthermore, there is nothing within Gosling or

Jagannathan that suggests validating variables and arguments of an application snapshot."

21.   Examiner respectfully disagrees. Jagannathan is specifically related to migrating a
process, including all relevant state data across heterogeneous machines. While Jagannathan
does not explicitly use the term 'snapshot' to describe the data structure used to migrate a
process, the term snapshot is well known in the art to describe a data structure that encapsulates
the objects, variables, and other state information related to a process, thread, or other executable
object at a specific point in time. As Jagannathan allows migration across machines using state
data encapsulated within this type of data structure, Jagannathan clearly meets the limitations of
"forming a snapshot of an executing program on a first computing device, transferring the
snapshot to a second computing device" (col. 18 lines 4-23; col. 19 lines 25-42; col. 20 lines 45-
55). Additionally, Jagannathan specifically states that for process migration to be used correctly
within an environment such as Java, verification procedures must be employed in order to
guarantee safe execution (col. 3 lines 17-49). Thus, the verification procedure of Gosling would
provide an exemplary model to allow the bytecode program to be safely analyzed and verified
for execution on the receiving processor.

Conclusion

22.   The prior art made of record and not relied upon is considered pertinent to applicant's

disclosure.

USPN **5,999,731**; USPN **6,052,732**; USPN **6,075,940**; USPN **6,247,171**; USPN **6,704,923**; USPN **6,477,702** are all related to the verification of bytecode programs to ensure type safety before execution on a processor.
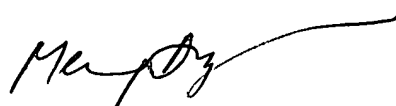
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
May 4, 2004

MENG-AL'T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100